DE HAAGSE HOGESCHOOL
FACULTEIT TECHNOLOGIE, INNOVATIE & SAMENLEVING

VOORBLAD SCHRIFTELIJKE TOETSEN

OPLEIDING	:	**ELEKTROTECHNIEK**
TOETSCODE	:	**UITWERKINGEN INLDIG**
GROEP	:	**EP1, EQ1D**
TOETSDATUM	:	**31 OKTOBER 2014**
TIJD	:	**13:00 - 14:30**
AANTAL PAGINA'S (incl. voorblad)	:	**10**
DEZE TOETS BESTAAT UIT	:	**6 open vragen**
GEBRUIK HULPMIDDELEN	:	**JA**
TOEGESTANE HULPMIDDELEN	:	**Aantekeningen, boeken, afdrukken van de PowerPoint-slides, programmeerbare/grafische rekenmachine**
TOETSOPGAVE INLEVEREN	:	**NEE**
OVERIGE OPMERKINGEN	:	**Beoordeling tentamen:** **Bij elke opgave staat het maximum aantal te behalen punten genoteerd, in totaal is maximaal 90 punten te behalen.** **Eindcijfer = 1 + (aantal behaalde punten / 10)**
OPSTELLER VAN DEZE TOETS	:	**J.E.J. op den Brouw**
TWEEDE LEZER VAN DEZE TOETS	:	**J.Z.M. Broeders**

BELANGRIJKSTE PUNTEN UIT DE TOETSREGELING VAN DE ONDERWIJS- EN EXAMENREGELING:

- Je dient je via Osiris ingeschreven te hebben voor deze toets.
- Schrijf je naam, je studentnummer, de toetscode en de naam van de docent meteen op het tentamenpapier.
- Leg je identiteitsbewijs op de hoek van de tafel.
- Zet alle elektronische communicatiemiddelen (mobiele telefoon, PDA, etc.) en horloges uit en stop deze in je tas; deze mogen niet als calculator of klok worden gebruikt.
- Je mag het lokaal het eerste halfuur van een toets niet verlaten.
- Volg de instructies op het toetsvoorblad.
- Steek je hand op als je een vraag hebt.

versie 1 oktober 2014 TEX

Let op: laat bij het beantwoorden van de vragen de berekening of uitwerking zien, antwoorden zonder berekening of uitwerking leveren geen punten op.

Opgave 1 (15 pt)
Gegeven de onderstaande twee getallen. De getallen staan in 2's complement notatie. Zet de getallen om naar het decimale talstelsel.

a) 11001101_2 (2 pt).

b) FACE_{16} (3 pt).

Een computerprogramma vraagt aan een gebruiker om een 8-bits unsigned getal in te voeren. De gebruiker voert het decimale getal 233 in. Door een fout in de programmatuur wordt het ingevoerde getal echter als 8-bits signed (2's complement) getal afgedrukt op het scherm.

c) Bereken welk decimaal getal het computerprogramma op het scherm afdrukt (3 pt).

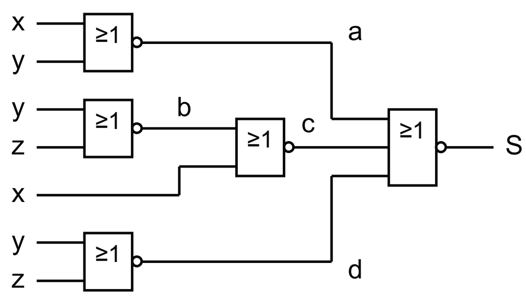
Gegeven onderstaand decimaal getal. Zet om naar een positief binair getal. Rond het antwoord af op 10 binaire cijfers achter de komma indien nodig.

d) $0{,}615_{10}$ (4 pt).

Binnenkort wordt een lander op komeet 67P/Churyumov-Gerasimenko gezet. De kleinste afstand van de komeet tot de zon (de zogenaamde *perihelion*) is 1,2429 Astronomische Eenheden. Dat komt overeen met 185.940.000 km.

e) Bereken hoeveel bits er minimaal nodig zijn om deze afstand van de komeet tot de zon uitgedrukt in km op te slaan in een geheel binair getal (3 pt).

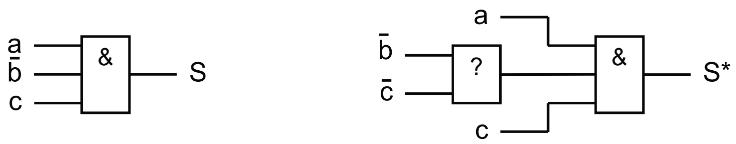
Opgave 2 (15 pt)
Gegeven de poortschakeling in figuur 1. De schakeling bestaat uit vijf NOR-poorten met ingangen x, y en z en uitgang S. In de figuur zijn de vier interne signalen a, b, c en d getekend.



Figuur 1: *Schakeling met NOR-poorten.*

a) Reken deze schakeling door en bepaal de logische functie in mintermvorm. Stel hiervoor een waarheidstabel op met daarin alle signalen in de figuur (10 pt).

Gegeven de twee schakelingen in figuur 2.



Figuur 2: *Twee equivalente schakelingen.*

De functies S en S^* moeten dezelfde logische werking hebben. Daarvoor kan een poort gekozen worden ter vervanging voor de poort met het vraagteken.

b) Welke logische functie moet de poort met het vraagteken vervullen om S en S^* dezelfde logische werking te laten hebben? Er kan gekozen worden uit: AND, OR of EXOR. Laat duidelijk zien hoe je aan het antwoord komt (5 pt).

Opgave 3 (15 pt)

a) Gegeven twee 8-bits 2's complement getallen 1110.1101_2 en 1101.1000_2. Tel deze twee getallen bij elkaar op. Laat duidelijk zien hoe de optelling verloopt en noteer de carry's (10 pt).

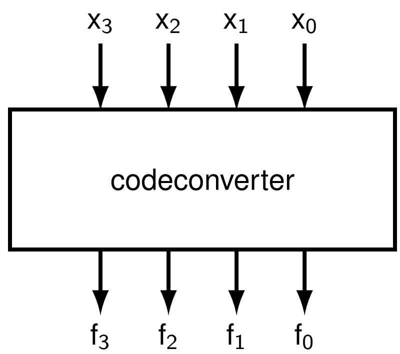
b) Is er sprake van overflow bij de optelling in a)? Motiveer het antwoord (5 pt).

Opgave 4 (25 pt)
In een bepaalde computer is een codeomzetter nodig voor de geïnverteerde Aiken-code. Hierbij worden de BCD-cijfers 0000 t/m 1001 (0 t/m 9) als volgt gecodeerd (omgezet):

- De cijfers 0 t/m 4 worden van 15 afgetrokken
- De cijfers 5 t/m 9 worden van 9 afgetrokken.

Merk op dat de bitcombinaties 1010 t/m 1111 niet gebruikt worden, want dat zijn geen geldige BCD-cijfers.

Voor deze omzetter is een schakeling nodig, zie figuur 3. De schakeling heeft als ingangen een 4-bits BCD-cijfer $X = x_3x_2x_1x_0$ en als uitgangen een 4-bits geïnverteerde Aiken-cijfer $F = f_3f_2f_1f_0$.



$F = 15 - X$ voor $0 \leq X \leq 4$
$F = 9 - X$ voor $5 \leq X \leq 9$

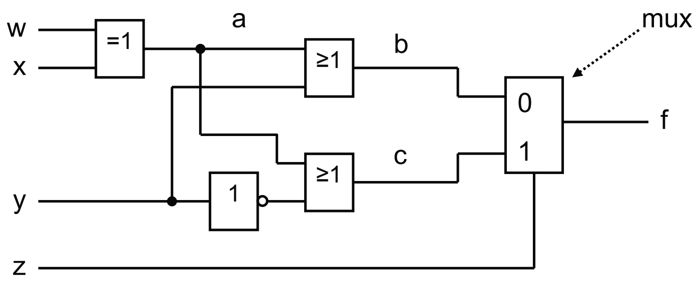
Figuur 3: *Codeerschakeling*

a) Geef de waarheidstabel voor uitgangen f_3, f_2, f_1 en f_0 van dit apparaat (5 pt).

b) Teken de Karnaughdiagrammen voor de in a) gevonden waarheidstabel en geef de vereenvoudigde functies in som-van-producten-vorm voor f_3, f_2, f_1 en f_0 (15 pt).

c) Teken een poortschakeling voor de in b) gevonden functies met alleen NOT-, AND-, OR- en EXOR-poorten (5 pt).

Opgave 5 (10 pt)
Een ontwerper van ARM heeft voor een nieuwe microprocessor een schakeling ontworpen, zie figuur 4. Zij is echter met de noorderzon vertrokken en heeft geen documentatie achtergelaten. Een nieuw team moet nu de logische werking en de vertragingstijden achterhalen. Daarvoor moet de schakeling doorgerekend worden. De vertragingstijden van de individuele poorten zijn gegeven in ns en opgenomen in tabel 1.



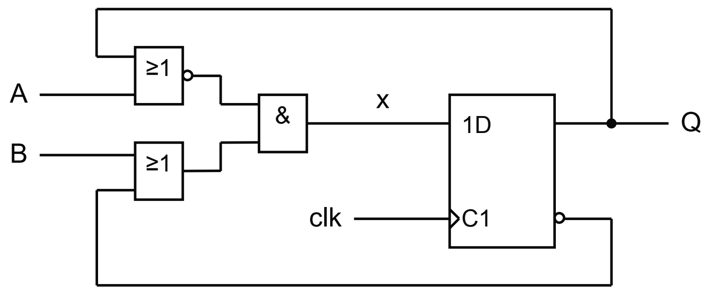
Figuur 4: *Deel microprocessorschakeling.*

Tabel 1: *Vertragingstijden*

Poort	$t_{P(min)}$	$t_{P(max)}$
EXOR	4,5	9,7
OR	3,1	7,2
NOT	1,5	4,3
Mux	5,2	8,3

Bepaal de minimale en maximale vertragingstijden van de gehele schakeling.

Opgave 6 (10 pt)
Gegeven onderstaande flipflopschakeling (figuur 5). Voor de D-ingang van de flipflop is een combinatorische schakeling geplaatst bestaande uit een NOR-, een OR- en een AND-poort.



Figuur 5: *Flipflopschakeling.*

De beginstand van de flipflop is 0 en signaal B is aan een logische 1 gekoppeld waardoor de functie van de combinatoriek wordt vereenvoudigd. Op ingang clk worden 9 klokpulsen aangeboden. Op ingang A wordt het volgende bitpatroon aangeboden: 011101001 (beginnend met de linker 0), elke (bit-)waarde steeds vlak voor de opgaande flank.

Geef bij elke (bit-)waarde van ingang A vlak voor de opgaande flank van ingang clk steeds de (bit-)waarde van uitgang Q vlak na de opgaande flank van ingang clk. Timing wordt buiten beschouwing gelaten.

o-o-o-o- einde toets -o-o-o-o-

Uitwerking opgave 1

(Opgave a) Het binaire getal 11001101_2 begint met een 1 dus het getal is negatief. Dat houdt in dat het meest significante bit, het zogenaamde tekenbit, als negatief moet worden beschouwd:

$$\begin{aligned}011001101_2 &= -1\cdot 2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 0\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0\\ &= -2^7 + 2^6 + 2^3 + 2^2 + 2^0\\ &= -128 + 64 + 8 + 4 + 1\\ &= -51_{10}\end{aligned}$$

Een alternatief is om het binaire getal eerst positief te maken (*flip bits, add one*) en dan om te zetten naar een decimaal getal. Zie slides week 5 pagina 25 en 26.

(Opgave b) Het hexadecimale getal FACE_{16} verdient wat meer aandacht. In feite is de hexadecimale schrijfwijze een notatie om binaire getallen verkort op te schrijven. Een hexadecimaal getal is dus eenvoudig als binair getal te schrijven door elk hexadecimaal cijfer weer te geven met precies vier bits: $\mathrm{FACE}_{16} = 1111.1010.1100.1110_2$. Nu is goed te zien dat het binaire getal begint met een 1, dus het betreft een negatief getal.

Het getal is direct naar het decimale talstelsel om te schrijven door het meest significante hexadecimale cijfer als negatief te zien: $\mathrm{F}_{16} = 1111_2 = \text{-}1_{10}$. Zie slides week 5 pagina 50 – 51.

$$\begin{aligned}\mathrm{FACE}_{16} &= -1\cdot 16^3 + 10\cdot 16^2 + 12\cdot 16^1 + 14\cdot 16^0\\ &= -4096 + 2560 + 192 + 14\\ &= -1330\end{aligned}$$

(Opgave c) Het getal 233_{10} moet omgerekend worden naar binair of hexadecimaal, dat levert 1110.1001_2 en $\mathrm{E9}_{16}$ op. Zie hieronder.

$$\begin{array}{rcrcrcl} 233 & \div 2 & = & 116 & \rightarrow & 1 \\ 116 & \div 2 & = & 58 & \rightarrow & 0 \\ 58 & \div 2 & = & 29 & \rightarrow & 0 \\ 29 & \div 2 & = & 14 & \rightarrow & 1 \\ 14 & \div 2 & = & 7 & \rightarrow & 0 \\ 7 & \div 2 & = & 3 & \rightarrow & 1 \\ 3 & \div 2 & = & 1 & \rightarrow & 1 \\ 1 & \div 2 & = & 0 & \rightarrow & 1 \\ \underline{0} & & & & & \end{array} \qquad \begin{array}{rcrcrcl} 233 & \div 16 & = & 14 & \rightarrow & 9 & \\ 14 & \div 16 & = & 0 & \rightarrow & 14 & \text{(E)} \\ \underline{0} & & & & & & \end{array}$$

Het binaire getal begint met een binaire 1, dus in 2's complement notatie betreft het hier een negatief getal. De hexadecimale variant is het snelst om te zetten:

$$\mathrm{E9}_{16} = -2\cdot 16 + 9 = -32 + 9 = -23$$

Het computerprogramma drukt het decimale getal -23 af.

Een alternatief is om uit te gaan van de definitie van een negatief 2's complement getal: het negatieve getal van B wordt genoteerd als $2^8 + B$ (dit alles mod 2^8) en dat is gelijk aan 233. Zie slides week 5 pagina 10. Bij dit 8-bits getal wordt dit dan

$$B \equiv 2^8 + B \equiv 233 \pmod{2^8} \quad \longrightarrow \quad B = -23$$

(Opgave d) Het omzetten geschiedt door herhaald vermenigvuldigen met twee en steeds het gehele deel te noteren en verder te gaan met de fractie. Hieronder de uitwerking:

0,615	x 2	=	1,23	$\rightarrow$	1
0,23	x 2	=	0,46	$\rightarrow$	0
0,46	x 2	=	0,92	$\rightarrow$	0
0,92	x 2	=	1,84	$\rightarrow$	1
0,84	x 2	=	1,68	$\rightarrow$	1
0,68	x 2	=	1,36	$\rightarrow$	1
0,36	x 2	=	0,72	$\rightarrow$	0
0,72	x 2	=	1,44	$\rightarrow$	1
0,44	x 2	=	0,88	$\rightarrow$	0
0,88	x 2	=	1,76	$\rightarrow$	1
0,76	x 2	=	1,52	$\rightarrow$	1
0,52	x 2	=	1,04	$\rightarrow$	1

0,04	x 2	=	0,08	$\rightarrow$	0
0,08	x 2	=	0,16	$\rightarrow$	0
0,16	x 2	=	0,32	$\rightarrow$	0
0,32	x 2	=	0,64	$\rightarrow$	0
0,64	x 2	=	1,28	$\rightarrow$	1
0,28	x 2	=	0,56	$\rightarrow$	0
0,56	x 2	=	1,12	$\rightarrow$	1
0,12	x 2	=	0,24	$\rightarrow$	0
0,24	x 2	=	0,48	$\rightarrow$	0
0,48	x 2	=	0,96	$\rightarrow$	0
0,96	x 2	=	1,92	$\rightarrow$	1
$\underline{0{,}92}$					

(Natuurlijk zijn de eerste 11 cijfers na de komma noodzakelijk, daarna kan gestopt worden). Dus $0,615_{10} = 0,10011101011100001010001\ldots_2$. De laatste 20 bits blijven zich herhalen. Dat houdt in dat $0,615_{10}$ dus niet exact kan worden weergegeven in het binaire systeem.

Voor afronding moet worden gekeken naar het eerste binaire cijfer na het aantal weer te geven binaire cijfers. Als dat een 1 is, dan moet naar boven worden afgerond, is het een 0 dan moet naar beneden worden afgerond. In dit geval is het een 1 (zie onderstreping), dus er moet naar boven worden afgerond. Zie slides week 2 pagina 23 en 26.

$$\begin{aligned} 0,45_{10} &= 0,1001110101|\underline{1}10\ldots_2 \\ &= 0,1001110110_2 \qquad \text{(na afronding)} \end{aligned}$$

(Opgave e) Het gaat hier om het aantal bits dat nodig is om een specifiek getal op te slaan, niet voor een reeks getallen met een aantal cijfers. Afstanden zijn altijd positief (of nul) dus is er geen tekenbit nodig. Het aantal bits dat nodig is om een positief getal op te slaan is

$$M \leq 2^n - 1$$

met n zo klein mogelijk. Met enig rekenwerk kan n expliciet gemaakt worden. Het aantal bits dat nodig is om deze afstand op te slaan is

$$n = \left\lceil \frac{\log(185.940.000 + 1)}{\log 2} \right\rceil = \lceil 27,47\ldots \rceil = 28$$

Er zijn minimaal 28 bits nodig om dit getal op te slaan. Zie slides week 2 pagina 32.

Uitwerking opgave 2

(Opgave a) De schakeling wordt doorgerekend en elke variabele wordt in een waarheidstabel geplaatst. Zie tabel 2. Merk op dat de signalen b en d identiek zijn.

Tabel 2: *Waarheidtabel voor functie S.*

x	y	z	a	b	c	d	S
0	0	0	1	1	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	0
0	1	1	0	0	1	0	0
1	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1
1	1	0	0	0	0	0	1
1	1	1	0	0	0	0	1

De functie is nu te schrijven in de mintermvorm

$$\begin{aligned} S &= x \cdot \overline{y} \cdot z + x \cdot y \cdot \overline{z} + x \cdot y \cdot z \\ &= m_5 + m_6 + m_7 \\ &= \sum m(5,6,7) \end{aligned}$$

(Opgave b) De functies S en S^* moeten dezelfde logische werking hebben, dus

$$S = S^* \quad \rightarrow \quad a \cdot \overline{b} \cdot c = a \cdot (\overline{b} ? \overline{c}) \cdot c$$

waarbij voor het vraagteken de functies AND, OR en EXOR onderzocht moeten worden.

Dit kan opgelost worden door voor elke mogelijke functie een waarheidstabel op te stellen, zie tabel 3. De tabel is snel op te stellen want bij alle combinaties waarbij $a = 0$ of $c = 0$ is, zijn de uitkomsten van de functies ook 0. Alleen de combinaties waarbij $a = 1$ en $c = 1$ moeten worden doorgerekend.

Tabel 3: *Waarheidstabel voor functie S en S*.*

a	b	c	$a \cdot \overline{b} \cdot c$	$a \cdot (\overline{b} \cdot \overline{c}) \cdot c$	$a \cdot (\overline{b} + \overline{c}) \cdot c$	$a \cdot (\overline{b} \oplus \overline{c}) \cdot c$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	0	0	0
1	1	1	0	0	0	0

Natuurlijk kan het ook met behulp van de schakelalgebra.

Voor de AND geldt $S^* = a \cdot (\overline{b} \cdot \overline{c}) \cdot c = a \cdot \overline{b} \cdot \overline{c} \cdot c = 0$ want $\overline{c} \cdot c = 0$. Hieruit volgt dat een AND-poort niet voldoet.

Voor de OR geldt $S^* = a \cdot (\overline{b} + \overline{c}) \cdot c = a \cdot \overline{b} \cdot c + a \cdot \overline{c} \cdot c = a \cdot \overline{b} \cdot c + 0 = a \cdot \overline{b} \cdot c$, een OR-poort voldoet.

De EXOR vergt wat meer werk. De term $\overline{b} \oplus \overline{c}$ is in feite een EXOR-poort waarbij beide ingangen geïnverteerd zijn zodat geldt:

$$S^* = a \cdot (\overline{b} \oplus \overline{c}) \cdot c = a \cdot (\overline{\overline{b}} \cdot \overline{c} + \overline{b} \cdot \overline{\overline{c}}) \cdot c = a \cdot b \cdot \overline{c} \cdot c + a \cdot \overline{b} \cdot c \cdot c = 0 + a \cdot \overline{b} \cdot c = a \cdot \overline{b} \cdot c$$

De EXOR-poort voldoet ook.

Resumerend kan gezegd worden dat de OR- en EXOR-poort voldoen aan de eis dat $S = S^*$.

Uitwerking opgave 3

(Opgave a) Het optellen van de twee getallen is niet zo lastig, ook al zijn het negatieve getallen. De normale optelprocedure kan gebruikt worden (daarom is het 2's complement ook bedacht). Naast de binaire getallen zijn ook de decimale equivalenten gegeven. Zie slides week 5 pagina 28 en 41.

1)11110000		carry's
11101101		-19
11011000	+	-40
1)11000101		-59

De uitgaande carry moet genegeerd worden, die is geen onderdeel van het antwoord. De carry is het resultaat van de wijze waarop het 2's complementsysteem werkt. Zie bv. slides week 5 pagina 16.

(Opgave b) Bij de optelling in a) is geen overflow opgetreden. Zowel de twee op te tellen getallen als het antwoord beginnen met een 1 en zijn negatief. Zie slides week 5 pagina 42.

Uitwerking opgave 4

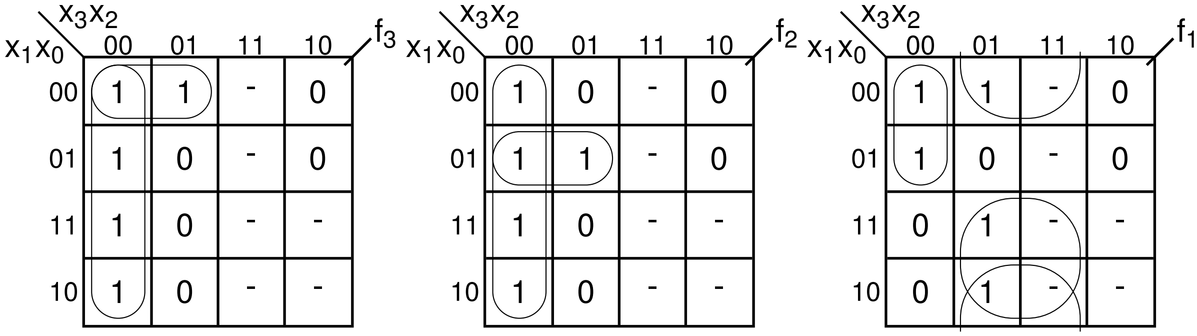
(Opgave a) Eerst moet de waarheidstabel worden opgezet. De bitcombinaties $X = x_3x_2x_1x_0$ kunnen gezien worden als een getal dat ligt tussen 0 en 15, waarvan alleen de combinaties 0 t/m 9 gebruikt worden, de overige komen niet voor. Dat worden don't cares in de waarheidstabel. De bitcombinaties $F = f_3f_2f_1f_0$ vormen de getallen $15 - X$ (voor $0 \leq X \leq 4$) of $9 - X$ (voor $5 \leq X \leq 9$).

Van alle mogelijkheden wordt een waarheidstabel opgezet, die is te vinden in tabel 4.

Tabel 4: *Waarheidstabel voor de functies f_3 t/m f_0.*

x_3	x_2	x_1	x_0	f_3	f_2	f_1	f_0
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	-	-	-	-
1	0	1	1	-	-	-	-
1	1	0	0	-	-	-	-
1	1	0	1	-	-	-	-
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

(Opgave b) Nadat de waarheidstabel is opgezet, moeten de functies uitgewerkt worden. Direct is te zien dat de functie voor f_0 erg eenvoudig is, namelijk $f_0 = \overline{x_0}$. De overige functies worden m.b.v. Karnaughdiagrammen uitgewerkt. Dit is weergegeven in figuur 6. Voor het opzetten en uitwerken van Karnaughdiagrammen zie slides week 3 pagina 40, 41 en 52.



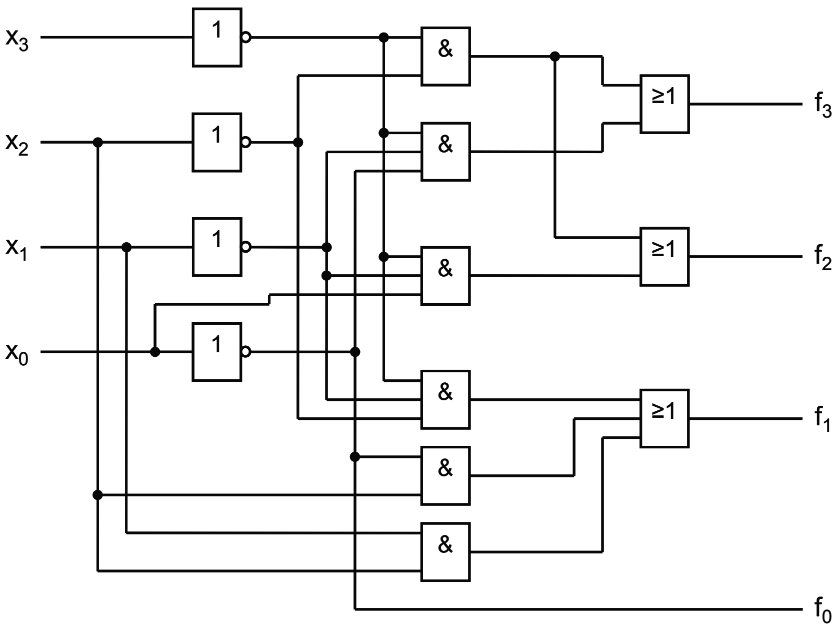
Figuur 6: *Karnaughdiagrammen voor de functies f_3 t/m f_1.*

De gevonden functies voor f_3 t/m f_0 zijn

$$f_3 = \overline{x_3} \cdot \overline{x_2} + \overline{x_3} \cdot \overline{x_1} \cdot \overline{x_0}$$
$$f_2 = \overline{x_3} \cdot \overline{x_2} + \overline{x_3} \cdot \overline{x_1} \cdot x_0$$
$$f_1 = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} + x_2 \cdot \overline{x_0} + x_2 \cdot x_1$$
$$f_0 = \overline{x_0}$$

Opmerkingen: De term $\overline{x_3} \cdot \overline{x_2}$ komt twee keer voor. In de functie van f_3 kan de term $\overline{x_3} \cdot \overline{x_1} \cdot \overline{x_0}$ vervangen worden door de term $x_2 \cdot \overline{x_1} \cdot \overline{x_0}$. In de functie van f_2 kan de term $\overline{x_3} \cdot \overline{x_1} \cdot x_0$ vervangen worden door de term $x_2 \cdot \overline{x_1} \cdot x_0$.

(Opgave c) Het schema voor de functies zijn weergegeven in figuur 7.



Figuur 7: *Schema voor de functies f_3 t/m f_0.*

Uitwerking opgave 5

Het berekenen van de minimale propagatietijd is niet zo lastig. Het kortste pad in tijd is vanaf ingang z via de multiplexer naar uitgang f. De minimale propagatietijd is dan

$$t_{P(min)}(f) = t_{P(min)}(\text{Mux}) = 5{,}2\,\text{ns}$$

Voor het berekenen van de maximale propagatietijd is wat meer werk nodig. Er moet nu gezocht worden naar het langste pad (in tijd). De ingangen y en z geven zeker niet de maximale tijd, hiervoor moet gestart worden met ingangen w en x. Signalen via w/x komen via een EXOR-poort aan bij interne knooppunt a. Daarna kan gekozen worden via intern knooppunt b en c. Beide leveren een even grote vertraging op. Als laatste moet de multiplexer nog worden gepasseerd. Het pad voor de maximale propagatietijd is dus $w/x \to a \to b/c \to f$:

$$\begin{aligned} t_{P(max)}(f) &= t_{P(max)}(\text{EXOR}) + t_{P(max)}(\text{OR}) + t_{P(max)}(\text{Mux}) \\ &= 9{,}7 + 7{,}2 + 8{,}3 = 25{,}2\,\text{ns} \end{aligned}$$

Uitwerking opgave 6

De D-ingang wordt aangestuurd door een combinatie van poorten. De logische functie van deze schakeling is

$$D^n = (\overline{A^n + Q^n}) \cdot (B^n + \overline{Q^n})$$

Nu wordt de B-ingang verbonden met een logische 1 zodat de functie zich vereenvoudigd naar een NOR-poort

$$B = 1 \quad \to \quad D^n = \overline{A^n + Q^n}$$

De flipflop heeft eerst de stand 0. De eerste waarde die aangeboden wordt is ook 0. Uit de tabel volgt dat D dus logisch 1 wordt. Zo moeten alle waarden van A worden afgegaan, steeds uitrekenen wat de waarde van Q wordt. Er wordt een tabel opgesteld met A^n, Q^n, D^n (is signaal x) en Q^{n+1}. Merk op dat $Q^{n+1} = D^n$ want dat is de werking van een D-flipflop. Zie tabel 5.

Tabel 5: *Doorlopen standen van de flipflop.*

nr.	A^n	Q^n	D^n	Q^{n+1}
1	0	0	1	1
2	1	1	0	0
3	1	0	0	0
4	1	0	0	0
5	0	0	1	1
6	1	1	0	0
7	0	0	1	1
8	0	1	0	0
9	1	0	0	0
10	-	0	-	-

o-o-o-o- einde uitwerkingen -o-o-o-o-